Weakly-Supervised Discovery of Named Entities Using Web Search Queries

Marius Paşca
Google Inc.
1600 Amphitheatre Parkway
Mountain View, California 94043
mars@google.com

ABSTRACT

A seed-based framework for textual information extraction allows for weakly supervised extraction of named entities from anonymized Web search queries. The extraction is guided by a small set of seed named entities, without any need for handcrafted extraction patterns or domain-specific knowledge, allowing for the acquisition of named entities pertaining to various classes of interest to Web search users. Inherently noisy search queries are shown to be a highly valuable, albeit little explored, resource for Web-based named entity discovery.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.2.7 [**Artificial Intelligence**]: Natural Language Processing; I.2.6 [**Artificial Intelligence**]: Learning; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

Named entities, knowledge acquisition, query logs, weakly supervised information extraction, unstructured text

1. INTRODUCTION

1.1 Motivation

Although the information in large textual collections such as the Web is available in the form of individual textual documents, the human knowledge encoded within the documents can be seen as a hidden, implicit Web of classes of instances (e.g., named entities), interconnected by relations applying to those instances (e.g., facts). The accurate identification of named entities pertaining to diverse classes is an essential step towards automatically constructing knowledge bases from unstructured text [16], particularly in the case of the acquisition of large fact repositories from Web documents [2]. Besides their role in populating larger repositories of human knowledge, named entities constitute a significant fraction of the queries submitted as part of an activity that permeates modern societies and is undertaken by millions of people every day, namely Web search. A variety of natural language processing tasks, including parsing, prepositional attachment [1] and coreference resolution [9], also benefit from the availability of information about named entities. Similarly, the performance of named entity recognizers improves when the systems have access to lists of accurate named entities [18], even with state of the art statistical recognizers [19]. Named entity discovery is also a powerful tool in building new verticals in Web search semi-automatically, for example to improve or provide alternative views of search results for popular classes such as *Drug*, *VideoGame* etc.

In a formal acknowledgment of the importance of named entities in Web search in particular, and text processing in general, numerous previous studies address the task of acquiring clean lists of named entities occurring within unstructured text. Traditionally, the recognition of names and their associated categories within unstructured text relies on manually compiled semantic lexicons and gazetteers. The amount of effort required to assemble large lexicons sometimes confines the recognition to a limited domain (e.g., *medical imaging*) for which large-coverage resources already exist. It is also possible to build recognizers that identify names automatically in text [3, 4, 18], although the targeted named entities are usually limited to a small set of coarse-grained classes, e.g. *Person* or *Location*. Recent work on large-scale information extraction holds much promise, as it scales well to Web-sized collections through an emphasis on lightweight extraction methods [2]. In particular, a few studies take advantage of non-traditional document collections for the purpose of mining named entities, in the form of comparable news articles [17] or multilingual corpora [6].

1.2 Contributions

In a departure from previous work on named entity discovery from unstructured text, this paper introduces a weakly supervised method for mining Web *search queries* in order to *explicitly* extract named entities that are expected to be relevant and suitable for later use. The method is inspired by recent work on weakly supervised acquisition of class attributes from query logs [12]. In contrast, previous work that looks at query logs as a data resource does so only to *implic-*

CIKM'07, November 6–8, 2007, Lisboa, Portugal.

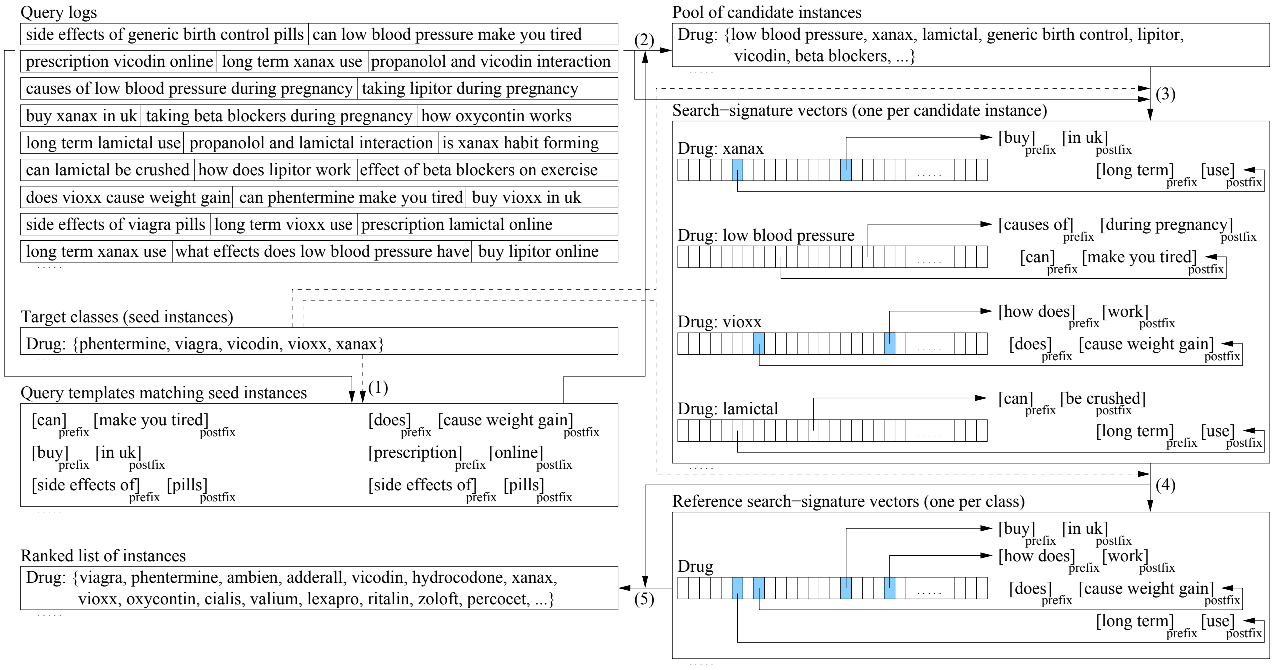
Query logs
side effects of generic birth control pills
can low blood pressure make you tired
prescription vicodin online
long term xanax use
propanolol and vicodin interaction
causes of low blood pressure during pregnancy
taking lipitor during pregnancy
buy xanax in uk
taking beta blockers during pregnancy
how oxycontin works
long term lamictal use
propanolol and lamictal interaction
is xanax habit forming
can lamictal be crushed
how does lipitor work
effect of beta blockers on exercise
does vioxx cause weight gain
can phentermine make you tired
buy vioxx in uk
side effects of viagra pills
long term vioxx use
prescription lamictal online
long term xanax use
what effects does low blood pressure have
buy lipitor online
.....
Target classes (seed instances)
Drug: {phentermine, viagra, vicodin, vioxx, xanax}
.....
Query templates matching seed instances
(1)
[can] prefix [make you tired] postfix
[does] prefix [cause weight gain] postfix
[buy] prefix [in uk] postfix
[prescription] prefix [online] postfix
[side effects of] prefix [pills] postfix
[side effects of] prefix [pills] postfix
.....
Ranked list of instances
Drug: {viagra, phentermine, ambien, adderall, vicodin, hydrocodone, xanax, vioxx, oxycontin, cialis, valium, lexapro, ritalin, zoloft, percocet, ...}
.....
(2)
Pool of candidate instances
Drug: {low blood pressure, xanax, lamictal, generic birth control, lipitor, vicodin, beta blockers, ...}
.....
(3)
Search−signature vectors (one per candidate instance)
Drug: xanax
[buy] prefix [in uk] postfix
[long term] prefix [use] postfix
Drug: low blood pressure
[causes of] prefix [during pregnancy] postfix
[can] prefix [make you tired] postfix
Drug: vioxx
[how does] prefix [work] postfix
[does] prefix [cause weight gain] postfix
Drug: lamictal
[can] prefix [be crushed] postfix
[long term] prefix [use] postfix
.....
(4)
Reference search−signature vectors (one per class)
Drug
[buy] prefix [in uk] postfix
[how does] prefix [work] postfix
[does] prefix [cause weight gain] postfix
[long term] prefix [use] postfix
.....
(5)

Figure 1: Overview of data flow during weakly supervised extraction of named entities from query logs

itly derive signals improving the quality of various tasks such as information retrieval, whether through re-ranking of the retrieved documents [21], query expansion [5], or the development of spelling correction models [8]. Conversely, previous studies in large-scale information extraction, including named entity discovery, uniformly choose to capitalize on document collections [10] rather than queries as preferred data source, thus failing to take advantage of the collective knowledge embedded haphazardly within noisy search queries.

The proposed extraction method is guided by a small set of seed instances, without any need for handcrafted extraction patterns or further domain-specific knowledge. To ensure varied experimentation on several dimensions despite the burden of manually assessing the correctness of the output, the evaluation computes the precision of instances extracted for ten different target classes (*City*, *Drug* etc.). The classes constitute a realistic sample of the wide range of domains of interest to Web search users. One of the prerequisites of the evaluation, illustrating its scope and time-intensive nature, is the manual assessment of the correctness of more than 4,000 candidate instances. The resulting precision scores are significant both in absolute value (0.96 for prec@50, 0.90 for prec@150, and 0.80 for prec@250), and relative to the quality of instances extracted from Web documents based on handcrafted patterns (with precision improving by 29% for prec@50, 26% for prec@150, and 15% for prec@250).

2. EXTRACTION FROM QUERY LOGS

2.1 Selection of Candidate Instances

Given a set of target classes and a set of seed instances for each class, the goal is to extract relevant class instances from query logs, without any further domain knowledge. As shown in Figure 1, the extraction method consists of five stages: identification of query templates that match the seed instances (Step 1); identification of candidate instances (Step 2); internal representation of candidate instances (Step 3) and seed instances (Step 4); and instance ranking (Step 5).

A target class (e.g., *Drug*), for which instances need to be extracted, is available in the form of a set of representative instances (e.g., *phentermine* and *vioxx*). In Step 1, each query that contains a seed instance generates a query template composed of the remainder of the query, that is, the prefix and the postfix around the matched instance. For example, the occurrence of the instance *vioxx* in the query *"long term vioxx use"* corresponds to the query template $[long\ term]_{prefix}$ $[use]_{postfix}$. During matching, all string comparisons are case-insensitive.

In another pass over the query logs, Step 2 collects a very large (and noisy) pool of candidate instances, by identifying the queries (e.g., *"long term xanax use"*) which match any of the query templates generated in the previous step, and collecting the query infixes (e.g., *xanax*) as candidate instances.

2.2 Representation through Search Signatures

Step 3 (see Figure 1) builds an internal representation of each candidate instance. Whenever a query contains a candidate instance collected in Step 2, the remainder of the query, that is, the concatenation of the remaining prefix and postfix, becomes an entry in a query template vector that acts as a *search signature* of the candidate instance with respect to the class. The weight of each entry in a vector is the frequency of occurrence of the contributing query in the query logs. For instance, the query *"can lamictal be crushed"* results in a new entry being added to the search-signature vector of *lamictal* with respect to the class *Drug*. The new entry corresponds to the query template $[can]_{prefix}$

Country:	*Drug*:	*VideoGame*:
what type of government does $\mathcal{I}$ have	how long does $\mathcal{I}$ stay in your system	how many copies of $\mathcal{I}$ have been sold
what do people in $\mathcal{I}$ eat	can $\mathcal{I}$ be crushed	how much does $\mathcal{I}$ cost
what is the weather like in $\mathcal{I}$ in march	what does the $\mathcal{I}$ pill look like	where can i play $\mathcal{I}$ online for free
how to apply for $\mathcal{I}$ visa	how much does $\mathcal{I}$ cost	how to install $\mathcal{I}$ mods
how did $\mathcal{I}$ gain independence	how does $\mathcal{I}$ affect the heart	when is $\mathcal{I}$ coming out on gamecube
where is $\mathcal{I}$ on the map	how is $\mathcal{I}$ manufactured	how many $\mathcal{I}$ levels
what continent is $\mathcal{I}$ on	does $\mathcal{I}$ cause weight gain	what copy protection does $\mathcal{I}$ use
what is $\mathcal{I}$'s currency	when was $\mathcal{I}$ fda approved	why does $\mathcal{I}$ crash
why did $\mathcal{I}$ join the eu	can $\mathcal{I}$ make you tired	how to add bots to $\mathcal{I}$ server
why is $\mathcal{I}$ poor	what $\mathcal{I}$ is made out of	who made $\mathcal{I}$ 2

Table 1: Examples of query templates from the reference search-signature vectors generated during named entity discovery ($\mathcal{I}$ stands for a class instance)

$[\textit{be crushed}]_{postfix}$. The weight of the new entry is set to the frequency of the query.

Step 4 of Figure 1 injects weak supervision in the extraction process, through the small input set of per-class instances. For each class (e.g., *Drug*), the vectors generated in Step 3 are inspected to identify those associated to a seed instance (e.g., *xanax* and *vioxx*), rather than any other candidate instance. The vectors thus selected are added together, forming a reference search-signature vector. Whereas Step 3 generates a vector for each candidate instance of each class, Step 4 produces one reference search-signature vector for each class. A reference vector can be thought of as a loose search fingerprint of the desired output type with respect to the class [12].

In Step 5, the similarity scores among the search-signature vector of each candidate instance, on one hand, and the reference search-signature vector of the class, on the other hand, determine the ordered list of candidate instances along with their ranking. The similarity scores are computed based on a similarity function, namely Jensen-Shannon, that is used frequently in text processing [7].

One of the possible interpretations of the reference search-signature vectors of a class is to view them as a series of queries (or questions, when formulated in natural language) that can be asked about the instances in the class. Table 1 shows some of the query templates in the reference search-signature vectors collected during named entity discovery for a sample of the target classes. Note how it is incrementally easier to guess to which class an instance belongs, solely by inspecting more questions that various people (users) ask about the instance. For example, if valid questions about an instance $\mathcal{I}$ include *"what type of government does $\mathcal{I}$ have"* and *"what do people in $\mathcal{I}$ eat"*, then it is relatively easy for people to guess that $\mathcal{I}$ must be a geopolitical entity of some sort, and later refine that guess as they are presented with more questions about the instance. The intuition behind named entity discovery from query logs is similar: given a set of candidate phrases, the system must guess which of the candidate phrases are more likely to belong to the target class, by looking at queries that can be asked about the candidate phrase, and more generally about all instances in the target class.

3. EXPERIMENTAL SETTING

3.1 Data

The input to the experiments is a random sample of around 50 million unique, fully-anonymized queries in English submitted by Web users to the Google search engine in 2006. All queries are considered independently of one another, whether they were submitted by the same user or different users, within the same or different search sessions.

3.2 Target Classes

To better test the ability of the extraction method to discover useful named entities, each target class is specified through only five seed instances, which are case-insensitive:

- *City*: {london, paris, san francisco, tokyo, toronto};
- *Country*: {argentina, china, costa rica, france, germany};
- *Drug*: {phentermine, viagra, vicodin, vioxx, xanax};
- *Food*: {chicken, fish, milk, tomatoes, wheat};
- *Location*: {argentina, china, france, san francisco, tokyo};
- *Movie*: {die hard, sin city, star trek, star wars, the matrix};
- *Newspaper*: {le monde, new york times, usa today, wall street journal, washington post};
- *Person*: {mel gibson, leonardo da vinci, jennifer lopez, pablo picasso, vincent van gogh};
- *University*: {columbia university, stanford university, university of chicago, university of pennsylvania, university of texas at austin};
- *VideoGame*: {grand theft auto, need for speed, quake, super mario bros., warcraft}.

The resulting set of ten target classes is a mix of coarse-grained classes (*Person* and *Location*) used heavily in literature on named entity discovery and extraction, and finer-grained classes (e.g., *Drug* and *VideoGame*) that more realistically model the inherent diversity of Web documents and queries. Furthermore, the classes differ from one another with respect to domains of interest (e.g., Health for *Drug* vs. Entertainment for *Movie*). Overall, the target classes provide varied experimentation on several dimensions, while taking into account the time intensive nature of manual accuracy judgments often required in the evaluation of information extraction systems [2].

3.3 Evaluation Procedure

Multiple lists of candidate instances (named entities) are evaluated for each class. The top 250 elements of each output list to be evaluated are sorted alphabetically into a merged list. Each instance of the merged list is manually assigned a correctness label with two possible values (*correct* or *incorrect*) within its respective class. Thus, a correctness label is manually assigned to a total of 4,468 instances extracted for the ten target classes, in a process that once again confirms that evaluation of information extraction methods can be quite time consuming.

Target Class	Extracted Instances				
	@10	@20	@250	@500	@750
City	ottawa	denver	minneapolis	tampa	virginia beach
Country	thailand	iceland	finland	europe	vietnam
Drug	cialis	effexor	concerta	lorazepam	zyrtec
Food	vegetable	lamb	cheese	ham	pumpkin
Location	singapore	new york city	dubai	montreal	amsterdam
Movie	superman	titanic	south park	smallville	fat albert
Newspaper	los angeles times	toronto sun	dallas morning news	sun sentinel	providence journal
Person	albert einstein	christopher columbus	rosa parks	theodore roosevelt	hitler
University	uc berkeley	university of georgia	university of oregon	auburn university	ucf
VideoGame	roller coaster tycoon	prince of persia	halo 2	red alert	hitman

Table 2: Sample of items at various ranks in the lists of instances acquired from query logs (Q$_{seed}$)

Target Class	Precision				
	@25	@50	@100	@150	@250
City	1.00	0.96	0.88	0.84	0.75
Country	1.00	0.98	0.95	0.82	0.60
Drug	1.00	1.00	0.96	0.92	0.75
Food	0.88	0.86	0.82	0.78	0.62
Location	1.00	1.00	1.00	1.00	1.00
Movie	0.92	0.90	0.88	0.84	0.79
Newspaper	0.96	0.98	0.93	0.86	0.54
Person	1.00	1.00	1.00	1.00	1.00
University	1.00	1.00	1.00	1.00	0.99
VideoGame	1.00	1.00	0.99	0.98	0.98
Average-Class	0.97	0.96	0.94	0.90	0.80

Table 3: Performance of named entity discovery from query logs (Q$_{seed}$)

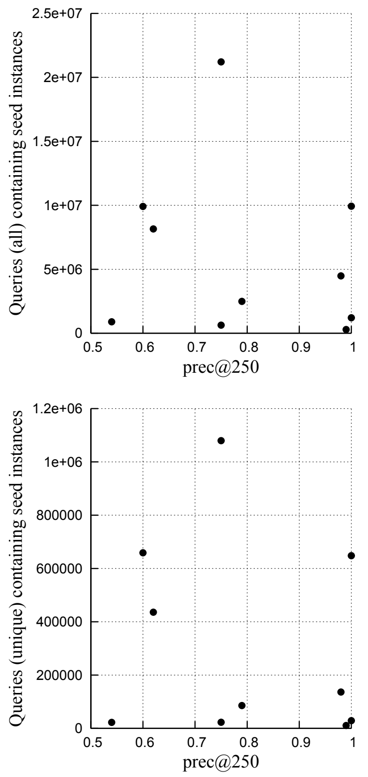

Figure 2: Correlation between the prec@250 score of a target class (see Table 3), and the corresponding number of all queries (top graph) and unique queries (bottom graph) containing any of the five seed instances of the target class

To compute the precision score over a ranked list of extracted instances, the correctness labels are converted to numeric values: *correct* to 1, and *incorrect* to 0. Precision at some rank N in the list is thus measured as the sum of the assigned values of the first N candidate instances, divided by N.

4. EVALUATION

4.1 Quality of the Extracted Instances

Using only the five seed instances per class and no further domain knowledge, the extraction method proposed in this paper (denoted Q$_{seed}$) mines the query logs for other useful instances, some of which are shown in Table 2. As expected, not all extracted instances are correct; for example, *europe* is an incorrect extraction in the case of the class *Country.* In a more rigorous analysis, Table 3 captures precision numbers at various ranks for each of the target classes. The top 250 instances extracted for both *Location* and *Person* contain no erroneous instances. For other classes, the accuracy of the extracted instances is quite good, at least at lower ranks (up to rank 50) regardless of the class. The variation in precision across different classes is more pronounced further down in the ranked lists of instances. In particular, the precision at rank 250 varies from a minimum of 0.54 for *Newspaper*, to a maximum of 1.00 for *Location* and *Person*, with a precision of 0.80 as an average over all target classes.

One may argue that the more queries contain any of the seed instances of a given target class, the better the internal representation (that is, reference search-signature vector) of that target class would be. In turn, better internal representations could lead to more accurate scoring, and hence to higher-quality lists of extracted instances. In other words, the extracted instances are more likely to be accurate, if the seed instances that define the class occur more frequently in the query logs. Figure 2 illustrates the correlation between the precision score of an individual target class, and the total number of all queries (top graph) or unique queries (bottom graph) that contain any of the five seed instances of the class. The corresponding correlation values between the precision of a class, on one hand, and the number of queries containing some seed instance of the class, on the other hand, is -0.17 when computed over all queries, or -0.19 when computed over unique queries. Both values show that, in fact, precision scores are not correlated with the popularity of the seed instances in the query logs. For instance, in the top graph of Figure 2, a total of more than 21 million queries contain seeds (*london*, *paris*, *san francisco*, *tokyo* or *toronto*) of the class *City*, whose list of extracted instances has prec@250 of 0.75. Comparatively, a lower number of queries, that is, around 4 million, contain seeds of the class *VideoGame*, yet the prec@250 score for that class is 0.98.

At ranks 250 and higher, lower precision numbers for classes such as *Country* are not necessarily meaningful, since there cannot be that many correct instances in those classes in the first place. On the other hand, we feel that it is not unrealistic to expect precision levels in excess of 0.90 at rank 250 and even higher, for the extraction method to be truly useful in accurately collecting large instance sets, in the order of thousands or even tens of thousands of instances (e.g., for *VideoGame* and *Movie*, not to mention *Person* and *Location*). With such an aggressive goal in mind, the current precision value of 0.80 at rank 250 is encouraging, but leaves room for improvement.

4.2 Comparison to Previous Results

A large body of previous work focuses on compiling lists of named entities exclusively from document collections, either iteratively by starting from a few seed extraction rules [3], or by mining named entities from comparable news articles [17] or from multilingual corpora [6]. In comparison, our paper is the first endeavor in named entity discovery from inherently-noisy Web search queries.

A set of extraction patterns relying on syntactically parsed text (e.g., <*subj*> *was kidnapped*) are acquired automatically from unstructured text in [14]. After manual post-filtering, the patterns extract relations in the terrorism domain (perpetrator, victim, target of a terrorist event) from a set of 100 annotated documents, with an average precision of 58%. A more sophisticated bootstrapping method [15] cautiously grows very small seed sets of five items, to less than 300 items after 50 consecutive iterations, with a final precision varying between 46% and 76% depending on the type of semantic lexicon. By adding the five best items extracted from 1700 text documents to the seed set after each iteration, 1000 semantic lexicon entries are collected after 200 iterations in [20], at precision between 4.5% and 82.9%, again as a function of the target semantic type.

In [2], handcrafted extraction patterns are applied to a collection of 60 million Web documents to extract instances of the classes *Company* and *Country*. Based on the manual evaluation of samples of extracted instances, the authors indicate that an estimated number of 1,116 instances of *Company* are extracted at a precision score of 0.90. Although these numbers are indicative of the quality of previous methods designed for Web documents, they are not directly comparable to the scores from Table 3 for two reasons. First, the numbers reported in [2] are computed from a sample of larger lists of extracted named entities, whereas the precision scores in Table 3 assume the systematic assessment of consecutive named entities from the extracted lists. Second, a detailed comparison of precision scores at various ranks cannot be obtained, unless the actual lists of named entities extracted from Web documents in previous studies are also already available, for each of the ten target classes. Naturally, this is not the case for [2], and in general for most previous work on named entity extraction from Web documents, since the authors report aggregated results on small, non-overlapping, pre-defined sets of target classes.

Target Class	Class Label (D_{patt})	
	Textual Label	Label Rank
City	cities	53
Country	countries	95
Drug	drugs	376
Food	foods	1159
Location	locations	44
Movie	movies	99
Newspaper	newspapers	172
Person	people	1
University	universities	132
VideoGame	video games	1532

Table 4: Manual one-to-one mappings from target classes to textual class labels collected from Web documents and deemed to be equivalent for the purpose of comparative evaluation

In an effort to comparatively assess the usefulness of query logs versus Web documents in the task of named entity discovery, the evaluation takes advantage of lists of instances as they were extracted in an earlier experiment on the acquisition of categorized named entities from Web documents [11]. As opposed to other previous work, the set of target classes in [11] is not specified in advance. Instead, it is incrementally acquired from Web documents along with the respective instances, based on handcrafted extraction patterns (denoted D_{patt}). For simplicity and similarly to previous studies on acquiring instances of arbitrary classes from textual documents [13, 2], the extraction patterns can be summarized as $\langle\mathcal{C}$ [such as|including] $\mathcal{I}\rangle$, where $\mathcal{I}$ is a candidate instance and $\mathcal{C}$ is the corresponding class label, as found in a document fragment that matches the patterns (e.g., *"[..] drugs such as Vioxx [..]"*).

A large number of class labels (e.g., *drugs*) are thus each associated with a ranked list of candidate instances, in the lists extracted with D_{patt} in [11]. As shown in the two leftmost columns of Table 4, it is trivial to map a class label derived in D_{patt} from Web documents, into exactly one target class deemed as lexically equivalent. Most mappings simply involve plural forms (e.g., *cities*) of the abstract target classes (e.g., *City*), with two exceptions dictated by the availability of a better mapping in the case of *Person* (mapped to *people* as a more natural alternative to the initial choice *persons*) and *VideoGame* (mapped logically to the phrase *video games*). The right column in Table 4 indicates the rank of each class label $\mathcal{C}$ among all class labels acquired in D_{patt}, when the ranking criterion is the frequency of occurrence of the class label $\mathcal{C}$ across all document fragments matching the extraction patterns. The values in Table 4 confirm that the chosen set of target classes provides varied experimentation, with ranks varying from 1 for *people* (corresponding to *Person*) to 1532 for *video games* (corresponding to *VideoGame*).

The instances extracted from Web documents (D_{patt}) are manually evaluated as *correct* or *incorrect*, following the same evaluation procedure used earlier for the instances acquired from query logs (Q_{seed}). The resulting precision scores exhibit a larger variation across target classes when compared to Q_{seed}. At rank 50, the minimum and maximum scores with D_{patt} are 0.30 and 0.98 (for *Food* and *Country*

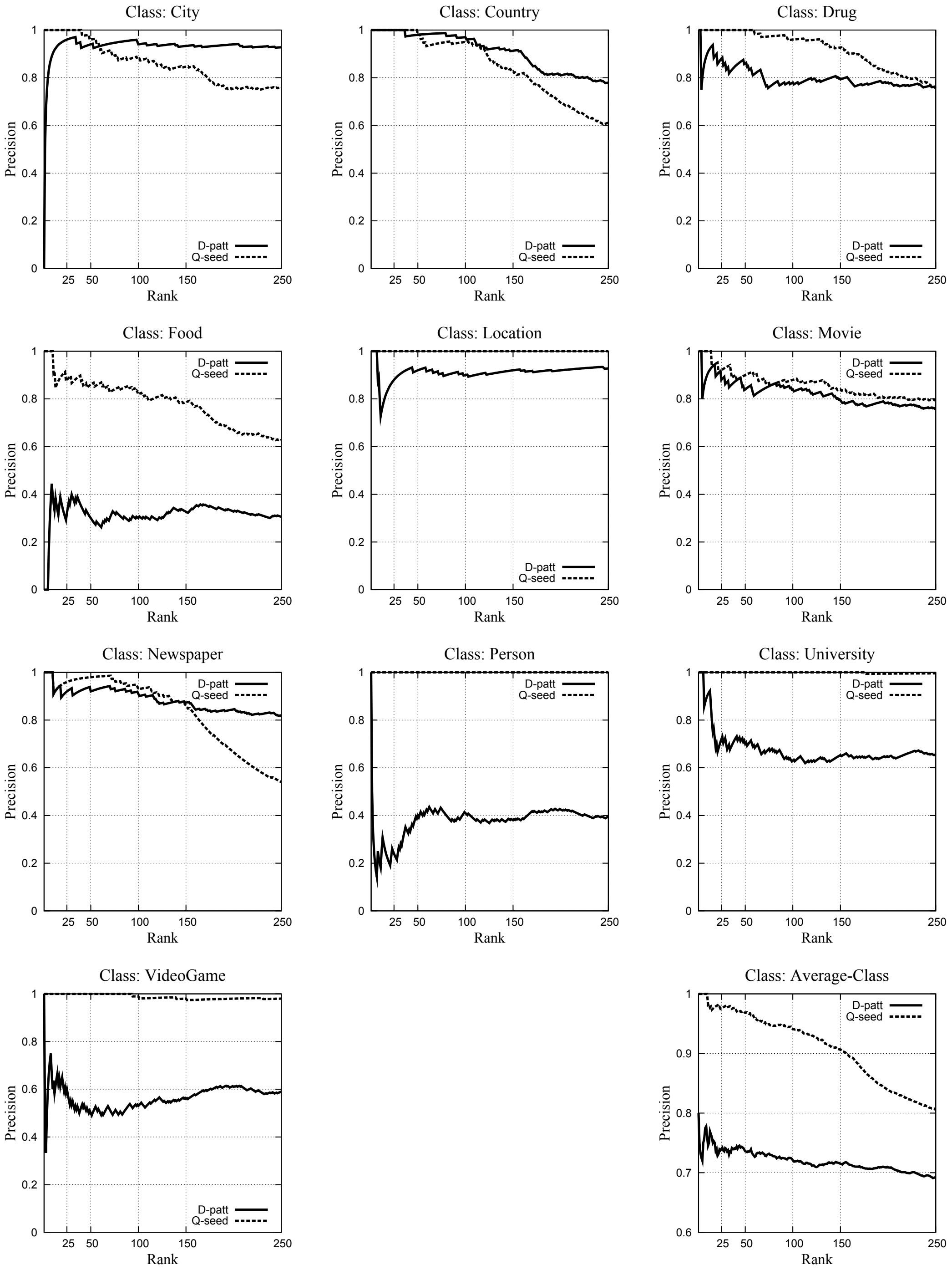

Figure 3: Extraction from Web documents with handcrafted patterns (D_{patt}), vs. seed-based extraction from queries (Q_{seed}) proposed in this paper, for individual target classes (first ten graphs) and as an average over all target classes (last graph)

respectively). In contrast, the scores at the same rank 50 vary between 0.86 and 1.00 in the case of Q_{seed}, as shown earlier in Table 3.

Figure 3 plots precision values for ranks 1 through 250 for pattern-based extraction from Web documents (D_{patt}) against seed-based extraction from search queries (Q_{seed}). The first ten graphs correspond to each of the ten target classes. The output of Q_{seed} is better for most target classes, with the exception of the classes *City* and *Newspaper* for which D_{patt} gives better precision at higher ranks, and *Country* for which D_{patt} is better across all ranks. The last graph in Figure 3 shows the comparative precision as an average over all target classes. Overall, the Q_{seed} method proposed in this paper outperforms D_{patt}, with relative precision score

Target Class	Top Instances	
City	D_{patt}	new york, los angeles, london, san francisco, dallas, boston, phoenix, cairo, chicago, seattle, beijing, atlanta, hollywood, miami, rome, detroit, toronto, athens, paris, jerusalem
	Q_{seed}	san francisco, london, paris, tokyo, toronto, new york city, washington dc, montreal, los angeles, ottawa, calgary, vancouver, san diego, edmonton, nyc, chicago, atlanta, philadelphia, winnipeg, denver
Country	D_{patt}	china, france, india, australia, japan, germany, canada, united states, brazil, russia, argentina, uk, mexico, thailand, belgium, usa, singapore, italy, spain, philippines
	Q_{seed}	france, china, germany, argentina, costa rica, spain, japan, italy, brazil, thailand, switzerland, sweden, poland, portugal, greece, new zealand, morocco, cuba, belgium, iceland
Drug	D_{patt}	ambien, viagra, prozac, lsd, nardil, imitrex, sonata, paxil, fioricet, esgic, epo, azt, gravol, metamucil, zoloft, valium, tumour necrosis factor, stemetil, rohypnol, hgh
	Q_{seed}	viagra, phentermine, vicodin, xanax, vioxx, ambien, adderall, hydrocodone, oxycontin, cialis, valium, lexapro, ritalin, zoloft, percocet, paxil, wellbutrin, tramadol, strattera, effexor
Food	D_{patt}	american, mexican, italian, chinese, fast food, popcorn, pop-tarts, seafood, hamburgers, kerala, japanese, french, steaks, asian, thai, indian, pizza, sandwiches, eggo, science diet
	Q_{seed}	fish, milk, chicken, wheat, tomatoes, shrimp, salmon, beef, pork, vegetable, vegetarian, meat, tuna, pasta, potato, rice, mushroom, lobster, tofu, lamb
Location	D_{patt}	france, gatwick airport, tamilnadu, ireland, new york, uk, mull, north, hungary, stanstead airports, south terminals, brisbane, pune, kerala, hyderabad, canada, philadelphia, dubrovnik airport, cork airport, basel airport
	Q_{seed}	france, china, san francisco, tokyo, argentina, japan, spain, italy, los angeles, singapore, hong kong, brazil, thailand, switzerland, new zealand, sweden, poland, germany, portugal, new york city
Movie	D_{patt}	rio bravo, rio lobo, mcclintock, el dorado, videos, star wars, titanic, lover, braveheart, swingers, natural born killers, forrest gump, striptease, short cuts, scream, scooby-doo, princess bride, jude, dvd, boogie nights
	Q_{seed}	star wars, die hard, star trek, the matrix, sin city, phantom of the opera, spiderman, harry potter, lord of the rings, superman, lotr, shrek, batman, scooby doo, scarface, inuyasha, finding nemo, naruto, godzilla, titanic
Newspaper	D_{patt}	new york times, casa grande dispatch, guardian, ath-thawra, washington post, wall street journal, independent, usa today, times, miami herald, aliraq, financial times, los angeles times, chicago tribune, athens news, sun sentinel, new york post, fulton telegraph, ar-riyadhi, ny times
	Q_{seed}	new york times, le monde, washington post, usa today, wall street journal, ny times, chicago tribune, boston globe, toronto star, los angeles times, houston chronicle, seattle times, vancouver sun, globe and mail, calgary herald, newsday, philadelphia inquirer, denver post, ottawa citizen, toronto sun
Person	D_{patt}	pasteur, i, hong kong, dr. weil, us, vin, babe ruth, you, yoko avs intermediates, salk, truffaut, martin luther king, christians, schroders, limp bizkit, sheep, god, flw, rockler woodworking, nirvana
	Q_{seed}	leonardo da vinci, rembrandt, andy warhol, pablo picasso, vincent van gogh, salvador dali, van gogh, frida kahlo, picasso, albert einstein, claude monet, michelangelo, adolf hitler, benjamin franklin, helen keller, thomas edison, abraham lincoln, harriet tubman, william shakespeare, christopher columbus
University	D_{patt}	harvard, stanford, michigan state university, oxford, mit, columbia, cambridge, arkansas-pine bluff, ucla, central michigan university, yale, heriot watt, berkeley, warwick, birmingham, texas a, seton hall, liverpool, leeds, university of michigan
	Q_{seed}	university of chicago, stanford university, university of texas at austin, columbia university, university of pennsylvania, boston university, northwestern university, university of virginia, university of florida, uc berkeley, university of miami, university of michigan, university of iowa, university of arizona, university of maryland, duke university, ohio state university, university of utah, university of minnesota, university of georgia
VideoGame	D_{patt}	final fantasy, macintosh, nintendo, mortal kombat, chrono trigger, doom, resident evil, street fighter, playstation, playstation, quake, palm, legend of zelda, dragon warrior, starcraft, zelda, mega man, tobal no., tomb raider, soul caliber
	Q_{seed}	grand theft auto, warcraft, need for speed, quake, super mario bros., gta, world of warcraft, doom, need for speed underground, roller coaster tycoon, age of empires, halo, half life, the sims, metal gear solid, tomb raider, half life 2, ghost recon, rome total war, prince of persia

Table 5: Top 20 items in the lists of instances acquired with handcrafted patterns from Web documents ($\mathbf{D}_{patt}$), vs. seed-based extraction from queries ($\mathbf{Q}_{seed}$) proposed in this paper

boosts of 29% (0.96 vs. 0.74) at rank 50, 26% (0.90 vs. 0.71) at rank 150, and 15% (0.80 vs. 0.69) at rank 250. Table 5 provides a more detailed view on the top items in the actual lists of instances extracted from Web documents and query logs respectively.

The extracted instances sometimes capture condensed variants that would otherwise be difficult to collect since they are rarely available in unstructured text, even within Web documents. For example, a variant such as *nfsmw*, which is actually extracted from search queries, may be less formal than its equivalent *Need for Speed: Most Wanted* mentioned on the official Web site of the game developer, but *nfsmw* is certainly quite popular among avid users searching for a *VideoGame*.

5. CONCLUSION

Traditional wisdom suggests that textual documents tend to assert information (statements or facts) about the world in the form of expository text. Comparatively, search queries can be thought of as being nothing more than noisy, keyword-based approximations of often-underspecified user information needs (interrogations). Despite this apparent disadvantage, and in a departure from previous approaches to large-scale acquisition of named entities from the Web, this paper illustrates the usefulness of a weakly-supervised extraction framework for mining named entities from query logs, rather than Web documents. Since the extracted instances relate to one another through various types of facts, they constitute a building block towards acquiring large sets of relations connecting named entities. Next steps include the acquisition of named entities for languages other than English and for target classes organized hierarchically (e.g., *Impressionist*, *Painter*, *Artist* and *Person*), as well as the exploitation of Web documents in general and Web search results in particular for additional extraction clues in conjunction with those derived from query logs.

6. REFERENCES

[1] E. Brill and P. Resnik. A transformation-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, pages 1198–1204, Kyoto, Japan, 1994.

[2] M. Cafarella, D. Downey, S. Soderland, and O. Etzioni. KnowItNow: Fast, scalable information extraction from the Web. In *Proceedings of the Human Language Technology Conference (HLT-EMNLP-05)*, pages 563–570, Vancouver, Canada, 2005.

[3] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the 1999 Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, pages 189–196, College Park, Maryland, 1999.

[4] S. Cucerzan and D. Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of the 1999 Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, pages 90–99, College Park, Maryland, 1999.

[5] H. Cui, J. Wen, J. Nie, and W. Ma. Probabilistic query expansion using query logs. In *Proceedings of the 11th World Wide Web Conference (WWW-02)*, pages 325–332, Honolulu, Hawaii, 2002.

[6] A. Klementiev and D. Roth. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 817–824, Sydney, Australia, 2006.

[7] L. Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL-99)*, pages 25–32, College Park, Maryland, 1999.

[8] M. Li, M. Zhu, Y. Zhang, and M. Zhou. Exploring distributional similarity based models for query spelling correction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 1025–1032, Sydney, Australia, 2006.

[9] K. McCarthy and W. Lehnert. Using decision trees for coreference resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1050–1055, Montreal, Quebec, 1995.

[10] R. Mooney and R. Bunescu. Mining knowledge from text using information extraction. *SIGKDD Explorations*, 7(1):3–10, 2005.

[11] M. Paşca. Acquisition of categorized named entities for Web search. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM-04)*, Washington, D.C., 2004.

[12] M. Paşca. Organizing and searching the World Wide Web of facts - step two: Harnessing the wisdom of the crowds. In *Proceedings of the 16th World Wide Web Conference (WWW-07)*, pages 101–110, Banff, Canada, 2007.

[13] P. Pantel and D. Ravichandran. Automatically labeling semantic classes. In *Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, pages 321–328, Boston, Massachusetts, 2004.

[14] E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049, Portland, Oregon, 1996.

[15] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99)*, pages 474–479, Orlando, Florida, 1999.

[16] L. Schubert. Turing's dream and the knowledge challenge. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, Boston, Massachusetts, 2006.

[17] Y. Shinyama and S. Sekine. Named entity discovery using comparable news articles. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, pages 848–853, Geneva, Switzerland, 2004.

[18] M. Stevenson and R. Gaizauskas. Using corpus-derived name lists for named entity recognition. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00)*, Seattle, Washington, 2000.

[19] P. Talukdar, T. Brants, M. Liberman, and F. Pereira. A context pattern induction method for named entity extraction. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, pages 141–148, New York, New York, 2006.

[20] M. Thelen and E. Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*, pages 214–221, Philadelphia, Pennsylvania, 2002.

[21] Z. Zhuang and S. Cucerzan. Re-ranking search results using query logs. In *Proceedings of the 15th International Conference on Information and Knowledge Management (CIKM-06)*, pages 860–861, Arlington, Virginia, 2006.